



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

reverse parse



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used reverse parse

Found 13,699 of 151,219

Sort results by [Save results to a Binder](#)[Try an Advanced Search](#)[Try this search in The ACM Guide](#)Display results [Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1 [Reversible unification based machine translation](#)**

Gertjan van Noord

August 1990 **Proceedings of the 13th conference on Computational linguistics - Volume 2**Full text available: pdf(580.98 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

In this paper it will be shown how unification grammars can be used to build a *reversible* machine translation system. Unification grammars are often used to define the relation between strings and meaning representations in a declarative way. Such grammars are sometimes used in a bidirectional way, thus the same grammar is used for both parsing and generation. In this paper I will show how to use bidirectional unification grammars to define *reversible* relations between language depe ...

2 [Bi-directional LR parsing from an anchor word for speech recognition](#)

Hiroaki Saito

August 1990 **Proceedings of the 13th conference on Computational linguistics - Volume 3**Full text available: pdf(488.54 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

This paper introduces a new technique of parsing sentences from an arbitrary word which is highly reliable or semantically important. This technique adopts an efficient LR parsing method and uses a *reverse LR table* constructed besides a standard LR table. This technique is particularly suitable in parsing a lattice of words hypothesized by a speech recognition module. If we choose anchor symbols in such a way that they are almost always acoustically reliable, the bi-directional LR parsing ...

3 [Semiring parsing](#)

Joshua Goodman

December 1999 **Computational Linguistics**, Volume 25 Issue 4
 Full text available: pdf(2.13 MB) [Publisher Site](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We synthesize work on parsing algorithms, deductive parsing, and the theory of algebra applied to formal languages into a general system for describing parsers. Each parser performs abstract computations using the operations of a semiring. The system allows a single, simple representation to be used for describing parsers that compute recognition, derivation forests, Viterbi, n-best, inside values, and other values, simply by substituting the operations of different semirings. We also show how t ...

4 Robust multilingual parsing using island grammars

Nikita Synytsky, James R. Cordy, Thomas R. Dean

October 2003 **Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  [pdf\(128.39 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Any attempt at automated software analysis or modification must be preceded by a comprehension step, i.e. parsing. This task, while often considered straightforward, can in fact be very challenging for some source code. Files that make up web applications serve as an example of such difficult-to-parse artifacts, for two reasons. First, these files often contain several programming languages at once, sometimes with widely varying syntaxes, and intermingled at the statement level. Second, the code ...

5 Leveraging IBM visual age for C++ for reverse engineering tasks

Johannes Martin

November 1999 **Proceedings of the 1999 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  [pdf\(283.87 KB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The IBM VisualAge for C++ development environment provides tool writers with the ability to query the compiler's internal data structures for information on the programs being compiled. This paper shows how these features can be used to write data extractors for supplying data to common reverse engineering tools, while significantly reducing the complexity of the data extractors and the time needed to develop these as compared to using traditional approaches and programming tools.

6 Modelling: Reveal: a tool to reverse engineer class diagrams

Sarah Matzko, Peter J. Clarke, Tanton H. Gibbs, Brian A. Malloy, James F. Power, Rosemary Monahan

February 2002 **Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications - Volume 10**

Full text available:  [pdf\(1.00 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Many systems are constructed without the use of modeling and visualization artifacts, due to constraints imposed by deadlines or a shortage of manpower. Nevertheless, such systems might profit from the visualization provided by diagrams to facilitate maintenance of the constructed system. In this paper, we present a tool, Reveal, to reverse engineer a class diagram from the C++ source code representation of the software. In Reveal, we remain faithful to the UML standard definition of a ...

Keywords: UML, automated construction, class diagram, object-oriented programming, reverse engineering, unified modeling language

7 Reverse engineering and system renovation—an annotated bibliography

M. G. J. van den Brand, P. Klint, C. Verhoef

January 1997 **ACM SIGSOFT Software Engineering Notes**, Volume 22 Issue 1

Full text available:  [pdf\(1.32 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

To facilitate research in the field of reverse engineering and system renovation we have compiled an annotated bibliography. We put the contributions not only in alphabetical order but also grouped by topic so that readers focusing on a certain topic can read their annotations in the alphabetical listing. We also compiled an annotated list of pointers to information about reverse engineering and system renovation that can be reached via


Internet. For the sake of ease we also incorporated a brief ...

Keywords: annotated bibliography, reverse engineering, system renovation

8 Reversible grammars and natural language processing

Gordon Franck

April 1992 **Proceedings of the 1992 ACM/SIGAPP Symposium on Applied computing: technological challenges of the 1990's**

Full text available:  pdf(821.10 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

9 Applications: Self-monitoring with reversible grammars

Günter Neumann, Gertjan van Noord

August 1992 **Proceedings of the 14th conference on Computational linguistics - Volume 2**


Full text available:  pdf(410.36 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

We describe a method and its implementation for self-monitoring during natural language generation. In situations of communication where the generation of ambiguous utterances should be avoided our method is able to compute an unambiguous utterance for a given semantic input. The proposed method is based on a very strict integration of parsing and generation. During the monitored generation step, a previously generated (possibly) ambiguous utterance is parsed and the obtained alternative derivat ...

10 Software evolution: Understanding software systems using reverse engineering technology perspectives from the Rigi project

Hausi A. Müller, Scott R. Tilley, Kenny Wong

October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1**

Full text available:  pdf(785.90 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

Software engineering research has focused mainly on software construction and has neglected software maintenance and evolution. Proposed is a shift in research from synthesis to analysis. Reverse engineering is introduced as a possible solution to program understanding and software analysis. Presented is reverse engineering technology developed as part of the Rigi project. The Rigi approach involves the identification of software artifacts in the subject system and the aggregation of these artif ...

Keywords: legacy software, program understanding, reverse engineering, software evolution

11 Information retrieval & extraction: Reverse queries in DATR

Hagen Langer


August 1994 **Proceedings of the 15th conference on Computational linguistics - Volume 2**

Full text available:  pdf(522.55 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

DATR is a declarative representation language for lexical information and as such, in principle, neutral with respect to particular processing strategies. Previous DATR compiler/interpreter systems support only one access strategy that closely resembles the set of inference rules of the procedural semantics of DATR (Evans & Gazdar 1989a). In this paper we present an alternative access strategy (*reverse query strategy*) for a non-trivial subset of DATR.

12 A List Structure Form of Grammars for Syntactic Analysis

Doron J. Cohen, C. C. Gotlieb

January 1970 **ACM Computing Surveys (CSUR)**, Volume 2 Issue 1Full text available:  pdf(1.25 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**13 Weak and Mixed Strategy Precedence Parsing**

A. V. Aho, P. J. Denning, J. D. Ullman

April 1972 **Journal of the ACM (JACM)**, Volume 19 Issue 2Full text available:  pdf(979.03 KB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**14 An extended theory of head-driven parsing**

Mark-Jan Nederhof, Giorgio Satta

June 1994 **Proceedings of the 32nd conference on Association for Computational Linguistics**Full text available:  pdf(756.70 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#) Publisher Site

We show that more head-driven parsing algorithms can be formulated than those occurring in the existing literature. These algorithms are inspired by a family of left-to-right parsing algorithms from a recent publication. We further introduce a more advanced notion of "head-driven parsing" which allows more detailed specification of the processing order of non-head elements in the right-hand side. We develop a parsing algorithm for this strategy, based on LR parsing techniques.

**15 Java implementation verification using reverse engineering**

David Cooper, Benjamin Khoo, Brian R. von Konsky, Michael Robey

January 2004 **Proceedings of the 27th conference on Australasian computer science - Volume 26**Full text available:  pdf(549.74 KB)Additional Information: [full citation](#), [abstract](#), [references](#)

An approach to system verification is described in which design artefacts produced during forward engineering are automatically compared to corresponding artefacts produced during reverse engineering. The goal is to automatically determine if an implementation is consistent with the original design. In the system described, XML Metadata Interchange (XMI) representations of Unified Modelling Language (UML) class diagrams are recovered from compiled Java class files. These are automatically compar ...

**16 LR recursive transition networks for Earley and Tomita parsing**

Mark Perlin


June 1991 **Proceedings of the 29th conference on Association for Computational Linguistics**Full text available:  pdf(652.69 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#) Publisher Site

Efficient syntactic and semantic parsing for ambiguous context-free languages are generally characterized as complex, specialized, highly formal algorithms. In fact, they are readily constructed from straightforward recursive transition networks (RTNs). In this paper, we introduce LR-RTNs, and then computationally motivate a uniform progression from basic LR parsing, to Earley's (chart) parsing, concluding with Tomita's parser. These apparently disparate algorithms are unified into a single impl ...



LR Parsing



A. V. Aho, S. C. Johnson

June 1974 **ACM Computing Surveys (CSUR)**, Volume 6 Issue 2Full text available:  [pdf\(1.85 MB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**18 Tree insertion grammar: a cubic-time, parsable formalism that lexicalizes context-free grammar without changing the trees produced**

Yves Schabes, Richard C. Waters

December 1995 , Volume 76 , 21 Issue 3 , 4

Full text available:

 [pdf\(2.31 MB\)](#) [Publisher Site](#)Additional Information: [full citation](#), [abstract](#), [references](#)

Tree insertion grammar (TIG) is a tree-based formalism that makes use of tree substitution and tree adjunction. TIG is related to tree adjoining grammar. However, the adjunction permitted in TIG is sufficiently restricted that TIGs only derive context-free languages and TIGs have the same cubic-time worst-case complexity bounds for recognition and parsing as context-free grammars. An efficient Earley-style parser for TIGs is presented. Any context-free grammar (CFG) can be converted into a lexica ...

19 Linear-time suffix parsing for deterministic languages

Mark-Jan Nederhof, Eberhard Bertsch

May 1996 **Journal of the ACM (JACM)**, Volume 43 Issue 3Full text available:  [pdf\(2.53 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)



We present a linear-time algorithm to decide for any fixed deterministic context-free language L and input string w whether w is a suffix of some string in L . In contrast to a previously published technique, the decision procedure may be extended to produce syntactic structures (pares) without an increase in time complexity. We also show how this algorithm may be applied to process incorrect input in linear time.

20 An efficient probabilistic context-free parsing algorithm that computes prefix probabilities

Andreas Stolcke

June 1995 **Computational Linguistics**, Volume 21 Issue 2

Full text available:

 [pdf\(2.30 MB\)](#) [Publisher Site](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We describe an extension of Earley's parser for stochastic context-free grammars that computes the following quantities given a stochastic context-free grammar and an input string: a) probabilities of successive prefixes being generated by the grammar; b) probabilities of substrings being generated by the nonterminals, including the entire string being generated by the grammar; c) most likely (Viterbi) parse of the string; d) posterior expected number of applications of each grammar production, ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:

[Adobe Acrobat](#)[QuickTime](#)[Windows Media Player](#)[Real Player](#)